UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/940,771	08/27/2001	Sergey A. Kryloff	009785-0130	8038

20572	7590	08/03/2004

GODFREY & KAHN S.C.
780 NORTH WATER STREET
MILWAUKEE, WI 53202

EXAMINER
STEELMAN, MARY J

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 08/03/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	09/940,771	KRYLOFF ET AL.
	Examiner	Art Unit	
	Mary J. Steelman	2122	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _8/27/2001, 12/26/2001, 9/9/2002_.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-20_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-20_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _27 August 2001_ is/are: a)☐ accepted or b)☒ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _12/26/2001_.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

1.   Claims 1-20 are pending.

Information Disclosure Statement

2.   IDS submitted 12/12/2001 has been considered.

Drawings

3.   The drawings are objected to because FIGs. 13 and 14 are informal drawings. Corrected

drawing sheets are required in reply to the Office action to avoid abandonment of the application.

Any amended replacement drawing sheet should include all of the figures appearing on the

immediate prior version of the sheet, even if only one figure is being amended. The figure or

figure number of an amended drawing should not be labeled as "amended." If a drawing figure

is to be canceled, the appropriate figure must be removed from the replacement sheet, and where

necessary, the remaining figures must be renumbered and appropriate changes made to the brief

description of the several views of the drawings for consistency. Additional replacement sheets

may be necessary to show the renumbering of the remaining figures. The replacement sheet(s)

should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to

obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the

applicant will be notified and informed of any required corrective action in the next Office

action. The objection to the drawings will not be held in abeyance.

Specification

4.   The use of the trademark PKZIP has been noted in this application. It should be

capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary

nature of the marks should be respected and every effort made to prevent their use in any manner

which might adversely affect their validity as trademarks.

5.   The abstract of the disclosure is objected to because it exceeds the 150-word limit.

Correction is required. See MPEP § 608.01(f).

Content of Specification

(j)   <u>Abstract of the Disclosure</u>: See MPEP § 608.01(f). A brief narrative of the
disclosure as a whole in a single paragraph of 150 words or less commencing on a
separate sheet following the claims. In an international application which has
entered the national stage (37 CFR 1.491(b)), the applicant need not submit an
abstract commencing on a separate sheet if an abstract was published with the
international application under PCT Article 21. The abstract that appears on the
cover page of the pamphlet published by the International Bureau (IB) of the
World Intellectual Property Organization (WIPO) is the abstract that will be used
by the USPTO. See MPEP § 1893.03(e).

6.   The title of the invention is not descriptive. A new title is required that is clearly

indicative of the invention to which the claims are directed.   A suggested title is "Software Patch

Generator Using Compression Techniques".

Claim Objections

7.   Claim 9, page 27, line 7 is objected to because of the following informalities: Recites

"write form the old version...", should be –write from the old version--. Change 'form' to

'from'. Appropriate correction is required.

Double Patenting

8.   Claims 1-3, 8, and 9 are rejected under the judicially created doctrine of double patenting

over claims 1 (maps to claims 1 and 8 of Application 09/940,177), 2 (maps to claim 2 of

Application 09/940177), 3, (maps to claim 3 of Application 09/940177), claim 4 (maps to claims

ℋ𝑎𝑟 1 and 8, of Application 09/940177), and claim 5 (maps to claim 9 of Application 09/940177) of

U.S. Patent No. 6,289,509 B1 to Kryloff, since the claims, if allowed, would improperly extend

the "right to exclude" already granted in the patent.

The subject matter claimed in the instant application is fully disclosed in the patent and is

covered by the patent since the patent and the application are claiming common subject matter,

as follows:

Instant Application: 09 / 940771

Claim 1.

A method of generating a patch file from an old version of data consisting of a

series of elements and a new version of data consisting of a series of elements where both

the old and new versions of data are stored in a memory and accessible by a data

processor, the method comprising the steps of:

establishing an alphabet comprising a plurality of letters for processing the old and new

versions of data wherein a word consists of a series of elements represented by one or

more letters of the alphabet;

sorting the old version of data with the data processor alphabetically according to the

established alphabet to create a first sorted list of words and maintaining a pointer for

each element of the first sorted list of words indicating the element' s original location in

the old version of data;

sorting the new version of data with the data processor alphabetically according to the

established alphabet to create a second sorted list of words and maintaining a pointer for

each element of the second sorted list of words indicating the element's original location

in the new version of data;

recursively comparing the first and second sorted lists of words one word at a time for a

match of the words;

upon finding a match of the words, searching the first and second sorted lists of words

again to find the largest sequence of coinciding elements preceding and succeeding the

match of words;

storing the largest sequence of coinciding elements in a coincidences list;

processing the coincidences list to remove duplicative coincidences creating a refined list

of coincidences;

sorting the refined list of coincidences by pointer in the new version data, creating a

sorted list of coincidences;

adding the pointer of each coincidence in the new version data to the largest sequence of

coinciding elements, creating an end block list;

creating a patch file from the sorted coincidences list and the end block list; and

compressing the patch file into a secure, portable compressed archive for distribution.

Per claim 2.

The method of claim 1 wherein the patch file includes a series of write commands.

Per claim 3.

The method of claim 2 wherein creating a patch file from the sorted coincidences

list and end block list includes the steps of:

(a) finding a largest block of coinciding elements of the sorted coincidences list that

matches the second sorted list of words and recording the location of the largest found

block;

(b) in the area before the previously found block, finding the next largest block of

coinciding elements of the sorted coincidences list that matches the second sorted list of

words and recording the location of the next largest found block',

(c) repeating step (b) until

(i) the first member in the sorted list of coincidences is reached; or

(ii) there are no matches between the second sorted list of words and the sorted

coincidences list;

(d) writing a write from the old version of data command and pointer and length

information to the patch file if the first member of the sorted list of coincidences is

reached; and

(e) writing a write from patch file command and length and patch information to the

patch file when there is no match between the second sorted list of data and the sorted

coincidences list.

Per claim 8.

A system for generating a patch file from an old version of data consisting of a

series of elements and a new version of data consisting of a series of elements, the system

comprising:

a data processor;

a memory storing the old and new versions of data;

means for sorting the old version of data with the data processor alphabetically according to an established alphabet to create a first sorted list of data and for maintaining a pointer for each element of the first sorted list of data indicating the element's original location in the old version of data;

means for sorting the new version of data with the data processor alphabetically according to an established alphabet to create a second sorted list of data and for maintaining a pointer for each element of the second sorted list of data indicating the element's original location in the new version of data;

means for searching the first and second sorted lists of data to find a match of data;

means for storing the match of data as a sequence of coinciding elements in a coincidences list;

upon finding a match of data, means for searching the first and second sorted lists of data again to find the largest sequence of coinciding elements preceding and succeeding the match of data;

means for storing the largest sequence of coinciding elements in the coincidences list;

means for processing the coincidences list to remove duplicative coincidences, creating a refined list of coincidences;

means for sorting the refined list of coincidences by pointer in the new version of data, creating a sorted list of coincidences;

means for adding the pointer of each coincidence in the new version of data to the largest sequence of coinciding elements, creating an end block list;

means for creating a patch file from the sorted coincidences list and end block list; and

means for compressing the patch file into a secure, portable compressed archive for distribution.

Per claim 9.

The system of claim 8 wherein the means for creating a patch file includes;

means for finding the largest block of coinciding elements of the sorted coincidences list that matches the second sorted list of data and recording the location of the largest found block of coinciding elements in a memory;

means for finding the next largest block of coinciding elements of the sorted coincidences list that matches the second sorted list of data and recording the location of the next largest block of coinciding elements in the memory;

means for writing a write form the old version of data command and pointer and length information to the patch file; and

means for writing a write from patch file command and length and patch information to the patch file.

In comparison to US Patent 6,289,509 B1 to Kryloff:

Per claim 1:

A method of generating a patch file from an old version of computer code consisting of a

series of elements and a new version of computer code consisting of a series of elements

where

both the old and new versions of computer code are stored in a memory and accessible by

a data processor, the method comprising the steps of:

establishing an alphabet for processing the old and new versions of computer code where

a word consists of one or more elements of the alphabet;

sorting the old version of computer code with the data processor alphabetically according

to the established alphabet to create a first sorted list of code and maintaining a pointer

for each element of the first sorted list of code indicating the element's original location

in the old version of computer code;

sorting the new version of computer code with the data processor alphabetically

according to the established alphabet to create a second sorted list of code and

maintaining a pointer for each element of the second sorted list of code indicating the

element' s original location in the new version of computer code;

recursively comparing the first and second sorted lists of code one word at a time for a

match of the codes;

upon finding a match of the codes searching the first and second sorted lists of code to

find the largest sequence of coinciding elements preceding and succeeding the match;

storing each sequence of coinciding words in a coincidences list;

processing the coincidences list to remove duplicative information; and

creating a patch file from the processed coincidences list.

Per claim 2.

The method of Claim 1, wherein the patch file includes a series of write commands.

Per claim 3.

The method of Claim 2, wherein creating a patch file from the processed coincidences list

includes the steps of (a) finding a largest block of coinciding elements of the processed

coincidences list that matches the second sorted list of code and recording the location of

the largest found block;

(b) in the area before the previously found segment, finding the next largest block of

coinciding elements of the processed coincidences list that matches the second sorted list

of code

and recording the location of the next largest block;

(c) repeating step (b) until

(i) the first member in the processed list of coincidences is reached; or

(ii) there are no matches between the second sorted list of code and the processed

coincidences list; .

(d) writing a write from the old version of computer code command and offset and length

information to the patch file if the first member of the processed list of coincidences is

reached;

and

(e) writing a write from patch tile command and length and patch information to the patch file when there is no match between the second sorted list of code and the processed coincidences list.

Per claim 4.

A system for generating a patch file from an old version of computer code consisting of a series of elements and a new version of computer code consisting of a series of elements the system comprising:

a data processor;

a memory storing the old and new versions of computer code;

means for sorting the old version of computer code with the data processor alphabetically according to an established alphabet to create a first sorted list of code and for maintaining a pointer for each element of the first sorted list of code indicating the element's original location in the first version of computer code;

means for sorting the new version of computer code with the data processor alphabetically according to an established alphabet to create a second sorted list of code and for maintaining a pointer for each element of the second sorted list of code indicating the element's original location in the new version of computer code;

means for searching the first and second sorted lists of code to find the sequence of coinciding elements preceding and succeeding the match;

means for storing the sequence of coinciding letters in a coincidences list;

means for processing the coincidences list to remove duplicative information

therefrom;

means for creating a patch file from the processed coincidences list.

Per claim 5.

A system as in Claim 4, wherein the means for creating a patch file includes:

means for finding the largest block of coinciding elements of the processed

coincidences list that matches the second sorted list of code and recording the location of

the largest found block of coinciding elements in a memory;

means for finding the next largest block of coinciding elements of the processed

coincidences list that matches the second sorted list of code and recording the location of

the next largest block of coinciding elements in the memory;

means for writing a write form the old version of computer code command and

offset and length information to the patch file; and

means for writing a write from patch file command and length and patch

information to the patch file.

Furthermore, there is no apparent reason why applicant was prevented from presenting

claims corresponding to those of the instant application during prosecution of the application

which matured into a patent. See *In re Schneller*, 397 F.2d 350, 158 USPQ 210 (CCPA 1968).

See also MPEP § 804.

Claim Rejections - 35 USC § 103

9.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

10.   Claims 1-4, 8-10, 14, 15, and 17 are rejected under 35 U.S.C. 103(a) as being

unpatentable over US Patent 5,832,520 to Miller, in view of US Patent 5,270,712 to Iyer, and

further in view of US Patent 4,939,639 to Lee.

Per claims 1 and 8:

A method (and system) of generating a patch file from an old version of data consisting of a

series of elements and a new version of data consisting of a series of elements where both the old

and new versions of data are stored in a memory and accessible by a data processor, the method

comprising the steps of:

Miller disclosed:

-recursively comparing the first and second sorted lists of words one word at a time for a

match of the words; (Miller: Col. 2, lines 50-51, "

-processing the coincidences list to remove duplicative coincidences creating a refined list of

coincidences; (Miller: Col. 2, lines 31, "

-creating a patch file from the sorted coincidences list and the end block list; and compressing the patch file into a secure, portable compressed archive for distribution. (Miller Col. 2, lines 40-46, "

Miller failed to disclose:

-establishing an alphabet comprising a plurality of letters for processing the old and new versions of data wherein a word consists of a series of elements represented by one or more letters of the alphabet;

However Iyer disclosed a 3-symbol source alphabet (Table 2, col. 12) to preserve sort order in compressed form.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Miller's invention, by incorporation Iyer's limitation using an alphabet to sort as this provides a cardinal ordering for a sort procedure.

The combination of Miller and Iyer does not disclose:

-sorting the old version of data with the data processor alphabetically according to the established alphabet to create a first sorted list of words and maintaining a pointer for each element of the first sorted list of words indicating the element' s original location in the old version of data;

-sorting the new version of data with the data processor alphabetically according to the established alphabet to create a second sorted list of words and maintaining a pointer for each element of the second sorted list of words indicating the element's original location in the new version of data;

However, Lee disclosed a method for alphabetical sorting (col. 4, lines 26-37) using a master file of the words to be sorted and a modification of the master file.

Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Miller / Iyer by including a second, new version because it would be used to create the patch file.

Miller disclosed:

-upon finding a match of the words, searching the first and second sorted lists of words again to find the largest sequence of coinciding elements preceding and succeeding the match of words; (Miller: Col. 8, lines 39-47, "…find strings to copy from the old file in order to create a reconstructed new file…When choosing between multiple matches on a new file string…")

-sorting the refined list of coincidences by pointer in the new version data, creating a sorted list of coincidences;

(Miller: Col. 3, lines 28-32, "The invention makes decisions about whether to copy or insert data by using a search algorithm that can do the following: use a 'current positions' pointer into the old file for copying data...)

-adding the pointer of each coincidence in the new version data to the largest sequence of coinciding elements, creating an end block list;

(Miller: Col. 9, lines 6-14, "Before the search function is called for the first time, pointers are initialized to the beginning of the old and new files...The X byte string at P_OLD in the old file is compared with the X byte string at P_NEW I the new file..." A hash table is formed for each word.)

Per claim 2:

-the patch file includes a series of write commands.

(Miller: Fig. 3, S12 – Build the raw diff file including a sequence of copy commands, insert commands and insertion strings.)

Per claims 3 and 9:

-(a) finding a largest block of coinciding elements of the sorted coincidences list that matches the second sorted list of words and recording the location of the largest found block;

(Miller: Fig. 5B, 60, Col. 9, lines 37-41, "If at step T55 one or more matching strings are found, a comparison of bytes past the initial match is conducted and the position and length of the longest matching string in the old file that matches the string in the new file is retained (T60)"))

-(b) in the area before the previously found block, finding the next largest block of

coinciding elements of the sorted coincidences list that matches the second sorted list of words

and recording the location of the next largest found block;

(Miller: Fig. 5B, Col. 8, line 39 – col. 10, line 64.)

-(c) repeating step (b) until

   (i) the first member in the sorted list of coincidences is reached, or

   (ii) there are no matches between the second sorted list of words and the sorted

coincidences list;

(Miller does not specifically disclose the repeating steps. However, this is part of a recursive

process in a well-known sorting algorithm. Thus, it would have been obvious to a person of

ordinary skill in the art at the time of the invention to repeat the process.

-(d) writing a write from the old version of data command and pointer and length information to

the patch file if the first member of the sorted list of coincidences is reached;

(Miller: Col. 2, lines 56-60.)

-(e) writing a write from patch file command and length and patch information to the patch file

when there is no match between the second sorted list of data and the sorted coincidences list.

(Miller: col. 2, lines 60-67.)

Per claims 4, 10, 14, 15, and 17:

-one or more patch files are compressed into the .ZIP tile format to decrease the storage and

transfer requirements of the patch files. (secure, portable compressed archive for distribution)

(Miller: Col. 3, lines 62-65, "The entire minimized difference file (minimized by the techniques mentioned above) may be finally compressed (sing a 'well-known' compression algorithm-like 'zip' – or proprietary compression technique) to reduce file size.")

11.   Claims 5-7, 11-13, 16, 18, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 5,832,520 to Miller, in view of US Patent 5,270,712 to Iyer, in view of US Patent 4,939,639 to Lee, and further in view of US Patent 6,049,671 to Slivka et al.

The combined references of Miller, Iyer and Lee disclosed generating patch files using an alphabetical sort. Miller disclosed a zip was beneficial for reduction of file size.
The combined references failed to disclose encryption, authentication, and digital signatures.
However Slivka disclosed:

Per claims 5, 11, 16, 19 and 20:
- .ZIP file further includes encryption and authentication using digital signatures to secure the contents of the patch files from unauthorized access and to validate the identity of the creator of the archive.
(Slivka: Col. 18, lines 15-22, "The ...function operates in two modes; a normal mode and verify only mode. The normal mode removes the digital signature, verifies it, and renames the self-extracting executable distribution file...", col. 18, lines 35-39, "If the digital signature is verified as correct, the extractor in the self-extracting executable distribution file extracts and decompresses...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention to modify the Miller, Iyer, Lee combination, by including details regarding

encryption, authentication, and digital signatures as provided by Slivka, because when

transferring patches, or update versions, these techniques ensure security.

Per claims 6 and 12:

- .ZIP file is in the format of a self-extracting Zip file.

(Slivka: Col. 18, lines 35-39, "If the digital signature is verified as correct, the extractor in the

self-extracting executable distribution file extracts and decompresses..." A .zip file is merely a

name of a type of compressed file.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention to modify the Miller, Iyer, Lee combination, by including details regarding self

extracting zip files, as provided by Slivka, because when transferring patches, or update versions,

a self-extracting feature makes installation easier.

Per claims 7, 13, and 18:

-the self-extracting .ZIP file includes a rules-based form of intelligence to detect the presence of

the appropriate files to be patched and to determine how the patching process should proceed.

(Any computer program includes some type of rules based form of intelligence. As an example

(Slivka, col. 13, lines 25-35), "SSD is used to distribute a self-extracting archive of files to a

client application...When this archive is executed, it extracts (using intelligence / rules) all of the

included files, and then executes one of those files to complete the installation automatically...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of

the invention to modify the Miller, Iyer, Lee combination, by including details regarding self

extracting zip files used in a patching process, as provided by Slivka, because when transferring

patches, or update versions, a self-extracting feature makes installation easier, as the patch can

automatically choose which files to install and where to install them, thereby reducing error.

Conclusion

12.   The prior art made of record and not relied upon is considered pertinent to applicant's

disclosure.

13.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The

examiner can normally be reached Monday through Thursday, from 7:00 A.M. to 5:30 P.M.   If

attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan

Dam can be reached on (703) 305-4552.

The fax phone number is (703) 872-9306 for regular communications and for After Final

communications. Any inquiry of a general nature or relating to the status of this application or

proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Mary Steelman

07/21/2004

ANTONY NGUYEN-BA
PRIMARY EXAMINER